IN THE CLAIMS:

Please amend the claims as follows:

1.   (Currently Amended)   A <u>computer-implemented</u> method of debugging code containing a user-specified breakpoint located within a <u>predetermined</u> region of the code, the method comprising:

<u>setting a machine recognizable entry point in the code;</u>

<u>setting a machine recognizable exit point in the code, wherein the entry point and the exit point define an entry and an exit, respectively, of the region;</u>

executing the code;

<u>entering the region by encountering the machine recognizable entry point during the execution;</u>

determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint<u>; exiting the region being determined by encountering the machine recognizable exit point during the execution</u>; and

if so, halting the execution of the code.

2.   (Currently Amended)   The method of claim 1, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and <u>wherein</u> execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.

3.   (Currently Amended)   The method of claim 1, further comprising:

encountering the user-specified breakpoint;

suspending the execution of the code at the user-specified breakpoint; and

in response to a user-specified run-to command received while execution of the code is suspended, executing the code until reaching ~~an~~ <u>the machine recognizable</u> exit point of the region ~~is reached~~.

4. (Currently Amended) The method of claim 1, wherein the user-specified breakpoint is a non-conditional breakpoint and <u>wherein</u> execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.

5. (Currently Amended) The method of claim 1, ~~further comprising, prior to determining~~ <u>wherein</u>:

<u>setting the machine recognizable entry point in the code comprises</u> setting an internal safety net entry breakpoint in the code ~~relative to an entry point of the region~~; and

<u>setting a machine recognizable exit point in the code comprises</u> setting a safety net exit breakpoint in the code ~~relative to an exit point of the region~~.

6. (Original) The method of claim 5, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

7. (Currently Amended) The method of claim 5, wherein setting the safety net exit <u>breakpoint</u> is performed automatically in response to encountering the internal safety net entry breakpoint.

8. (Currently Amended) The method of claim 5, wherein the <u>machine recognizable</u> entry point and the <u>machine recognizable</u> exit point are determined by a compiler.

9. (Currently Amended) The method of claim 1, ~~wherein the region of the code is defined by an entry point and an exit point and~~ wherein halting comprises encountering a safety net breakpoint located in the code relative to the <u>machine recognizable</u> exit point.

Page 3

358266_2

10.   (Currently Amended)   The method of claim [[9]] 1, ~~further comprising determining, by a compiler,~~ wherein the machine recognizable entry point and the machine recognizable exit point ~~prior to executing the code~~ are determined by a compiler.

11.   (Currently Amended)   A computer readable medium containing a debug program which, when executed, performs an operation for debugging code containing a user-specified breakpoint located within a predetermined region of the code, the operation comprising:

defining the region of the code by a machine recognizable entry point and a machine recognizable exit point;

executing the code;

entering the region during the execution;

determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint; exiting the region being determined by encountering the machine recognizable exit point during the execution; and

if so, halting the execution of the code.

12.   (Currently Amended)   The computer readable medium of claim 11, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and wherein execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.

13.   (Currently Amended)   The computer readable medium of claim 11, wherein the user-specified breakpoint is a non-conditional breakpoint and wherein execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.

14.   (Currently Amended)   The computer readable medium of claim 11, wherein ~~the operation further comprises, prior to halting the execution of the code~~ defining the region comprises:

Page 4

358266_2

setting an internal safety net entry breakpoint in the code relative to ~~an~~ the machine recognizable entry point of the region; and

setting a safety net exit breakpoint in the code relative to ~~an~~ the machine recognizable exit point of the region.

15.   (Original)   The computer readable medium of claim 14, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

16.   (Currently Amended)   The computer readable medium of claim 14, wherein setting the safety net exit breakpoint is performed automatically in response to encountering the internal safety net entry breakpoint.

17.   (Currently Amended)   The computer readable medium of claim 14, wherein the machine recognizable entry point and the machine recognizable exit point are determined by a compiler.

18.   (Currently Amended)   The computer readable medium of claim 11, ~~wherein the region of the code is defined by an entry point and an exit point and~~ wherein halting comprises encountering a safety net breakpoint located in the code relative to the machine recognizable exit point.

19.   (Currently Amended)   The computer readable medium of claim [[18]] 11, wherein the machine recognizable entry point and the machine recognizable exit point are determined by a compiler.

20.   (Currently Amended)   A computer system, comprising:

a memory containing at least a debug program and code containing a user-specified breakpoint located within a predetermined region of the code; and

a processor which, when executing content of the memory, is configured to perform an operation comprising:

Page 5

358266_2

defining the region of the code by a machine recognizable entry point and a machine recognizable exit point;

executing the code;

determining whether the execution of the code exits the region of the code without firing the user-specified breakpoint; and

if so, halting the execution of the code upon reaching the machine recognizable exit point of the region.

21.   (Currently Amended)   The computer system of claim 20, wherein the user-specified breakpoint is a conditional breakpoint having an associated condition and wherein execution exits the region without firing the user-specified breakpoint because the associated condition is not satisfied.

22.   (Currently Amended)   The computer system of claim 20, wherein the user-specified breakpoint is a non-conditional breakpoint and wherein execution exits the region without firing the user-specified breakpoint because the user-specified breakpoint is not encountered.

23.   (Currently Amended)   The computer system of claim 20, wherein the operation further comprises, prior to ~~halting the execution of the code~~ determining whether the execution exits the code without firing the user-defined breakpoint:

setting an internal safety net entry breakpoint in the code relative to ~~an~~ the machine recognizable entry point of the region; and

setting a safety net exit breakpoint in the code relative to ~~an~~ the machine recognizable exit point of the region.

24.   (Original)   The computer system of claim 23, wherein setting the internal safety net entry breakpoint and setting the safety net exit breakpoint are performed automatically in response to a user selection of the region.

Page 6

358266_2

25. (Currently Amended)   The computer system of claim 23, wherein setting the safety net exit <u>break</u>point is performed automatically in response to encountering the internal safety net entry breakpoint.

26. (Currently Amended)   The computer system of claim 23, wherein the <u>machine recognizable</u> entry point and the <u>machine recognizable</u> exit point are determined by a compiler.

27. (Currently Amended)   The computer system of claim 20, ~~wherein the region of the code is defined by an entry point and an exit point and~~ wherein halting comprises encountering a safety net breakpoint located in the code relative to the <u>machine recognizable</u> exit point.

28. (Currently Amended)   The computer system of claim [[27]] <u>20</u>, wherein the <u>machine recognizable</u> entry point and the <u>machine recognizable</u> exit point are determined by a compiler.

358266_2